Reading

22. Wednesday 9 Mar: Sethi, Section 3.3, 4.1, 4.9
23. Friday 11 Mar: Sethi, Sections 4.2; Gentle Intro, Section 2.1; Davie, Section 4.7
24. Monday 21 Mar: Sethi, Section 4.6
25. Wednesday 23 Mar: Gentle Intro, Section 10.3

Exercises

Due date: Tuesday 22 Mar. Please leave your paper in the Taylor Hall homework box (it's located between 2.132 and 2.136) **by 4pm**.

1. [12] Give the result of performing the substitutions in each of the following expressions:

a. $\left(a\times b\div c\right)^{c,\ a,\ b}_{a\times y,\ b+c,\ b-3}$

b. $\left(a+b\times c\right)^{c,a,b}_{b,c,a}$

c. $\left(\left(\left(a+b\times c\right)^{c}_{b}\right)^{a}_{c}\right)^{b}_{a}$

2. [5] Find the values of the variables after the execution of *each* of the two statements in the following sequence:

$x,\ y,\ z := 3,\ 4,\ 5;$

$x,\ y,\ z := z+2,\ x-1,\ x*y$

3. [12] What are the possible final values of x after the execution of each of the following selection statements?

a. $x := 3$; **if** $x < 4 \rightarrow x := 2$ ▯ $x > 4 \rightarrow x := 3$ **fi**

b. $x := 4$; **if** $x < 4 \rightarrow x := 2$ ▯ $x > 4 \rightarrow x := 3$ **fi**

c. $x := 3$; **if** $x < 5 \rightarrow x := 2$ ▯ $x > 3 \rightarrow x := 3$ **fi**

d. $x := 4$; **if** $x < 5 \rightarrow x := 2$ ▯ $x > 3 \rightarrow x := 3$ **fi**

4. [6] List the successive states (i.e., the values of x and y after each iteration) of this program:

$x,\ y := 36,\ 44;$
do $x < y \rightarrow y := y - x$
▯ $x > y \rightarrow x := x - y$
od

5. [10] Sethi, Exercise 3.1 Instead of rewriting the conditionals as Sethi suggests, translate them into Guarded Commands. Each answer should be a single command, with as many alternatives as necessary.

6. [20] Determine (wp S R) for each of the following pairs S, R:

a. $S =$ "$x := 6 - y \cdot 2$", $R = (x = 14)$

b. $S =$ "$x := y - x$", $R = (x > -4 \cdot y)$

c. $S =$ "$i := i + 2\ ;\ j := i - j$", $R = (i = j)$

d. $S =$ "$i, j := 1 - i,\ j - i$", $R = (i = j)$

7. [10] Find the weakest precondition Q such that the following holds:

$$\{ Q \} \; x, y := y - 4, 3 \cdot x \; \{ x < 3 \cdot y \}$$

8. [15] Assuming that x and y are integers, use the method of weakest preconditions to determine whether

$$\{ x = 12 \wedge y = 4 \} \; x,y := x+y, x-6; \; x := y+8 \; \{ x+y > 4 \}$$

9. [10] Sethi, Exercise 3.1.

10. [10] Rewrite your answers to Exercise 9 in Guarded Command Notation.

11. [10] Sethi, Exercise 3.2. Instead of drawing flow diagrams, rewrite the program fragments in GCN.

12. [25] Sethi, Exercise 3.6. Develop a program for binary search from the description in Example 3.4, on page 81.

Develop the binary-search program in GCN. You should include the loop invariant and the program's precondition and postcondition, but you may develop the program informally.

13. [15] Sethi, Exercise 3.9. Instead of using `goto` statements, translate the fragments into GCN. Assume that in part (a), `i+1` should be `i-1`.

14. [10] Determine whether a type error is detected for the following statement in various different situations:

```
if a > 0 and a < 10
then a := sqrt(true);
else a := a / a;
end
```

a. Static type checking is performed, `a := 0` initially, and evaluation of **and** expressions is short-circuit.

b. Static type checking is performed, `a := 9` initially, and evaluation of **and** expressions is not short-circuit.

c. Dynamic type checking is performed, `a := 0` initially, and evaluation of **and** expressions is short-circuit.

d. Dynamic type checking is performed, `a := 9` initially, and evaluation of **and** expressions is not short-circuit.

Parts (e) and (f) refer to the following statement:

```
if a > 0 and sqrt(true) < 0
then a := a + 1;
else a := a _ 1;
end
```

e. Dynamic type checking is performed, `a := 0` initially, and evaluation of **and** expressions is short-circuit.

f. Dynamic type checking is performed, `a := 9` initially, and evaluation of **and** expressions is not short-circuit.

15. [15] Following the example of the type inference for `foldr` shown in Slide 163, determine the types of the following functions.

Of course you can check your answers by using Hugs or looking in Prelude.hs, so what's important in this exercise is to show the steps by which you determine these types.

a.
```
foldl f z []      = z
foldl f z (x:xs)  = foldl f (f z x) xs
```

b.
```
zipWith z (a:as) (b:bs)   = z a b : zipWith z as bs
zipWith _ _      _        = []
```

c.
```
takeWhile p []    = []
takeWhile p (x:xs)
      | p x       = x : takeWhile p xs
      | otherwise = []
```

CS 345 Richards

Assignment 6

Authors:

1. ____________________

2. ____________________

3. ____________________

4. ____________________